Literature Review: Reducing the Costs of Knowledge Acquisition in Word Sense Disambiguation, by Julie King

April 4, 2014

INTRODUCTION

Word Sense Disambiguation (WSD) is a part of the field of Natural Language Processing (NLP) and is the process of identifying which sense of a word is correct given the word's context [1]. For example, "paper" can have several meanings: a thin, flat sheet usually made of fibers; an essay; a newspaper; currency; hanging posters or leaflets; or hanging wallpaper. Based on the context of the word, humans can usually easily discern which meaning is appropriate. Getting a computer program to do this is not an easy task, but it is necessary for applications that seek to emulate a human's approach to language.

WSD is a complex task that depends on a vast amount of knowledge [1]. Knowledge about the multiple meanings of words and how to distinguish them must be collected and imparted to a program. Dictionaries, thesauruses, ontologies, collections of text (corpora), etc. are all sources of knowledge for WSD [1]. Some of this knowledge has been collected and standardized into resources heavily used by WSD researchers: WordNet, a collection of words arranged in sets of synonyms (synsets), and SemCor, an annotated corpus containing information about words' senses and parts of speech, are two of the most popular resources [1]. While external knowledge sources such as WordNet and SemCor can help cut some costs by eliminating the need for duplication of effort, they do not contain all possible knowledge regarding word senses, and tweaking or supplementation may be necessary for particular situations [1].

Manual involvement in WSD knowledge acquisition contributes to the high costs of WSD [1], [2]. Human hours are expensive, and manual identification of word senses and manual supervision and tagging of training examples, while effective for producing high quality WSD results, are also very time consuming [1], [2]. Indeed, the cost of manual knowledge acquisition in WSD is such a significant problem that it has been given a name: the knowledge acquisition bottleneck [1].

Problems with the costs of knowledge acquisition have kept WSD from being of good practical use in NLP in real-life applications so far [1]. Finding ways to either reduce the need for or reduce the costs of manual word sense generation and manual supervision and tagging of training examples should significantly help reduce costs of WSD in general. Much research is being done on finding WSD methods, particularly ones involving automation, that can reduce these costs. Following are some examples of recent approaches.

IMPORTANT IDEAS

Generating data about word senses, either to supplement or replace sources like WordNet, sometimes requires manual creation of knowledge by a human being via hand labeling words with their word sense(s) [1], [2]. Not only is this time-consuming, it is also tedious. Seemakurty, Chu, von Ahn, and Tomasic [2] took an entertainment-based approach to trying to solve the problem of the cost of developing data sets of word senses, with the premise that if the process were less tedious, it would take less time and thus be less costly. The authors attempted to transform the tedium involved with manual labeling of training data for WSD by turning the labeling process into a game. Initially, the authors used a multiple-choice style quiz as the game, with possible word senses for the sample word as the list of choices. The game contained ten

words and multiple sentences containing the words. There were problems with random guessing, which introduced no useful data; limiting the player's decision about the word sense to the list of provided suggestions, which may not contain the answer the player would give if prompted to come up with one of their own; and tedium, because players had to read multiple dictionary definitions. Consequently, the authors changed the game so that the players entered synonyms of their own choice for a presented term or phrase, and players earned points if their answer matched that of their partner. Also, the faster the entry, the more points were given. The authors noted that one of the interesting facets of this approach is that the synonyms were produced by the general public rather than trained lexicographers, which may give more insight into how words are actually used in daily life rather than straight out of a dictionary. Indeed, some of the senses generated by the players were both correct and not in WordNet. The error rate (senses that did not really match with the words) was significant, however. The authors concluded that the game is a fun and sustainable way to address the overhead involved with labeling training data but needs refinement to produce higher quality results.

As mentioned previously, knowledge acquisition costs can be necessary for sense generation beyond the senses contained in sources like WordNet. External knowledge sources can fall short when presented with text from particular fields, or domains, where the use of many words is slightly or substantially different than standard use [3]. In such cases, those general sources are not as efficient for certain situations as a program based on the particular text at hand could be. Manual labeling is always a possibility, but automation can be far less costly. Pantel and Lin [3] studied the use of a clustering algorithm they hoped would help show possible meanings of words for a given text that might be overlooked by a standard resource such as WordNet, as well as eliminate unnecessary consideration of inapplicable rare meanings

contained in such sources. The authors created a word sense discovery algorithm called "Clustering By Committee" (CBC) that discovers clusters, or "committees," within a text and assigns words to similar clusters, resulting in each of the clusters a word belongs to being one of that word's senses. The algorithm had some difficulty with words that could be more than one type of word (e.g., a noun that can also be a verb). Overall, however, the clustering algorithm resulted in clusters of words that contained fewer duplicate senses and discovery of infrequent senses of words that might nonetheless be important in the given context.

Identification of word senses is not limited to single words. A significant problem encountered in WSD is the unique challenge of noun compounds, such as "hot dog," "grey matter," or "washing machine." The meaning of a compound often cannot be determined simply by looking at the meaning of each individual noun component [4]. Consequently, conventional WSD methods often stumble with noun compounds [4]. Problems like this add to the already significant costs of conventional WSD methods.

Kim and Baldwin [4] attempted to disambiguate noun compounds by examining the word sense of the compound components as they relate to each other, using automated WSD techniques. The authors don't dispute that context is important in determining the meaning of a noun compound, but as interpreting context is a very difficult task, the authors chose to try to determine the word sense of the noun compounds out of context. The authors looked at the role of a noun within a compound to help determine its meaning. For instance, is the word used as a modifier or the "head noun"? The sense distributions of nouns (the most frequent meanings) varied with the role of the noun, with certain senses being more prevalent with certain roles. Using TiMBL, a memory-based learner that classifies based on the k-nearest neighbor approach, the authors studied two classification methods: supervised sense collocation and unsupervised

lexical substitution. The supervised method involved an equation using the grammatical role of the target noun, the semantic relation, and the semantics of the non-target noun. The semantic relation between the nouns was identified using 20 different possible relations; examples of these relations are shown in Figure 1. The unsupervised method involved lexical substitution: the target noun was replaced by its possible synonyms (taken from WordNet), and a probability score was assigned to the result based on the frequency of the noun compound with the substitute word in a given body of work; the word sense of the substitute compound with the highest frequency was chosen as the word sense for the original noun compound. The supervised classifier performed the best. With that method, the grammatical role of the target noun was the most influential factor. Accuracy based on the grammatical role declined when information about the semantic relation was added, suggesting that grammatical role is more important than semantic relation when determining the word sense of the target noun. The unsupervised method performed poorly compared to the supervised method. The authors concluded that WSD techniques are very helpful in interpretation of noun compounds and that this conclusion could lead to more research regarding noun compound interpretation.

Relation	**Definition**	**Example**
agent	n2 is performed by n1	student protest, band concert, military assault
content	n1 contains n2	paper tray, eviction notice, oil pan
located	n1 is located at n2	building site, home town, solar system
possessor	n1 has n2	student loan, company car
result	n1 is a result of n2	storm cloud, cold virus, death penalty

Fig. 1. Some semantic relations used by Kim and Baldwin, where n1 is the head noun and n2 is the modifying noun; these examples are quoted from Fig. 1 in [4].

Constructing databases for WSD through a supervised system that relies on manually tagged examples to help determine context is costly because the tagging involves a tremendous amount of manual work [5]. Also, even after tagging is done, searching these large databases for similar examples to the one being disambiguated takes significant time [5]. Fuji, Tokunaga, Inui, and Tanaka [5] approached this problem by asking whether samples can be used for example-based word sense disambiguation systems to cut down overhead costs without sacrificing quality. The authors used an example-based verb sense disambiguation system in a corpus-based approach that contained example sentences for each verb sense. The program calculated the similarity between the target sentence and the example sentences (a nearest-neighbor approach), then chose the verb sense with the highest similarity score. Prior research showed that this kind of system works well when there are many manually supervised examples in general and many supervised examples for each verb. However, the overhead for this manual supervision is huge. The authors proposed cutting costs by using a subset of examples for training. The subset of examples used to train the system was chosen based on maximum training utility. Training utility was determined by the increase the example would provide in interpretation certainty. An example with no close neighbors already in the database would thus be preferable to an example with a close neighbor, because the information in the example with a close neighbor is already partially covered by the close neighbor. The results were compared with example subsets created based on random selection, uncertainty sampling, and committee-based selection. The training utility sampling outperformed the random and committee-based sampling, cutting down on the number of examples needed for a given level of accuracy. With regard to uncertainty sampling, training utility sampling performed substantially better with smaller data sizes yet

worse with large data sizes. In general, the authors showed their sampling method reduces supervision and searching overhead yet preserves performance.

CONCLUSION

Recent research indicates that there are multiple viable (or possibly viable) ways to decrease costs associated with manual involvement in WSD knowledge acquisition. Seemakurty, Chu, von Ahn, and Tomasic's game-based approach [2] seems to liven up the task of generating word senses, but I wonder whether the approach would be suitable for a large database of words. With such a small sample of words, it is difficult to tell how the approach would work when scaled up to a database of hundreds or thousands of words, particularly given the mediocre quality of the results with the small sample of words. Also, the authors claim this is a low-cost method, but they do not provide any empirical results of time (and thus cost) savings, so it is not clear whether the approach is indeed an effective cost-cutting method. Pantel and Lim [3] demonstrated that replacing manual human work with automated algorithms is an effective means of reducing costs for discovering word senses, as well as fine-tuning word senses for particular corpuses. Use of a clustering algorithm like the one studied by Pantel and Lim [3] could be particularly helpful in alleviating additional costs of sense generation (or elimination) necessary for domain-specific work in which words senses are likely to be highly dependent on the domain, making a general resource like WordNet less effective than one that could be generated by something like their CBC algorithm. Costs can also be reduced by automating WSD for particular subsets of word types, such as noun compounds, as shown by Kim and Baldwin [4]. Using WSD techniques on a subset problem like this ends up helping the overall effectiveness of WSD by generating additional knowledge. Being able to automatically generate

word senses for compound nouns is a significant cost reduction over having a human manually list meanings for the compounds, and this method could perhaps be extended to other complex parts of speech. Finally, selective sampling based on training utility, as in Fuji, Tokunaga, Inui, and Tanaka's work [5], can help to lower costs of using supervised training examples, without diminishing quality of results. While their work was based on verb senses, the authors believe that their method is likely to be applicable to and useful for other example-based systems [5], thus increasing the potential cost savings of such a method across many WSD approaches.

WSD can contribute significantly to the success of NLP, but only if it is also cost-effective. WSD currently suffers from the high costs of knowledge acquisition. Helping to solve the costs of WSD and improve its effectiveness could help NLP performance in important areas: machine translation, information retrieval (more precise results with a focus on results with words with the appropriate senses and elimination of results with words with inapplicable senses), analysis and categorization, information extraction, information generation, word processing (more effective spelling and grammar checking), etc. [1], [2], [3], [5], [6]. If WSD is more cost-friendly and effective, more useful and possibly commercially successful applications involving NLP can thrive. Current research suggests that eventually WSD will be more cost effective, and thus more useful in practice.

REFERENCES

[1] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv*., vol. 41, no. 2, pp. 10:1 – 10:69, Feb. 2009.

[2] N. Seemakurty, J. Chu, L. von Ahn, and A. Tomasic, " Word sense disambiguation via human computation," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 2010, pp. 60-63.

[3] P. Pantel and F. Lin "Discovering word senses from text," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 613-619.

[4] S. N. Kim and T. Baldwin, "Word sense and semantic relations in noun compounds," *ACM Trans. Speech Lang. Process*., vol. 10, no. 3, pp. 1-17, July 2013.

[5] A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka, "Selective sampling for example-based word sense disambiguation," *Comput. Linguist*., vol. 24, no. 4, pp. 573-597, Dec. 1998.

[6] K. W. Church and L. F. Rau, "Commercial applications of natural language processing," *Commun. ACM*, vol. 38, no. 11, pp. 71-79, Nov. 1995.